

Multi-Level Fixed-Outline Floorplanning Using Convex Optimization

Houra Banihashem

Aminollah Mahabadi

Department of Electrical Engineering, Shahed University, Tehran, Iran

Abstract

In this paper, we present a novel multi-level fixed-outline convex-optimization floorplanning framework with different objectives on each level of global optimization of wirelength and chip area satisfactions. We solved the framework with a multi-Pass SA-based method for accurate floorplan result generation and significant reduction of the floorplanner running time. The framework started with a clustering method try to classify the modules as a best proposed initialization of the planning for connection-length minimizing. Next, an attractor-repeller model provides the relative positions of the floorplan blocks for wire-length minimizing. Finally, overlap-free and min deadspace floorplans are achieved in a fixed-outline with any specified percentage of whitespace for overlapped-length minimizing. The experimental results on the standard benchmarks GSRC and MCNC demonstrate none-overlap floorplans to minimize the wirelength, deadspace and area with significant improvements of run time under complete area utilization as the number of blocks increases.

Keywords: Multi-Level Floorplanning, Fixed-Outlined, Convex Modeling, Simulated Annealing (SA).

1. Introduction

Future generation of technology enables designers with millions of components being placed on a single chip to achieve high performance results. Due to the increasingly high complexity of designing modern chips, there is a requirement for design automation VLSI tools to produce high system performance with a reasonable consuming time. The computational complexity of the most problems in layout design is NP-hard [13]. The future works of VLSI physical circuit design will rely on the development of automation tools for all design phases, especially on fixed-outline floorplanning [8].

Floorplanning problem aims to arrange a set of rectangular modules on a fixed chip area to consider chip design so as to optimize an appropriate measure of performance [29]. This is an NP-hard problem and particularly challenging for the fixed-outline chip dimension that increasingly important as a tool to reduce time-to-market of chip product [25].

1.1. Floorplanning Design Process

The rapid growth in the complexity, size and the density of VLSI circuits has made floorplanning challenging that impact on the performance of the chip layout [8]. The goal of floorplanning design is to physically realize the modules obtained from its logic and functional description represented by the circuit diagram (Net list). Naturally, an application graph (e.g. directed acyclic graph) as a circuit diagram is the input to the floorplan design [20].

The layout of the floorplanning process is arranging a set of rectangular modules on a rectangular chip area to optimize an appropriate measure of performance. This problem could be more efficiently solved if it could be formulated as a convex model. The resulting layout is called an optimal floor plan. However, NP hardness of this process needs to present a fast reasonable running time with other properties such as scalability, efficiency, effectiveness and so on [29].

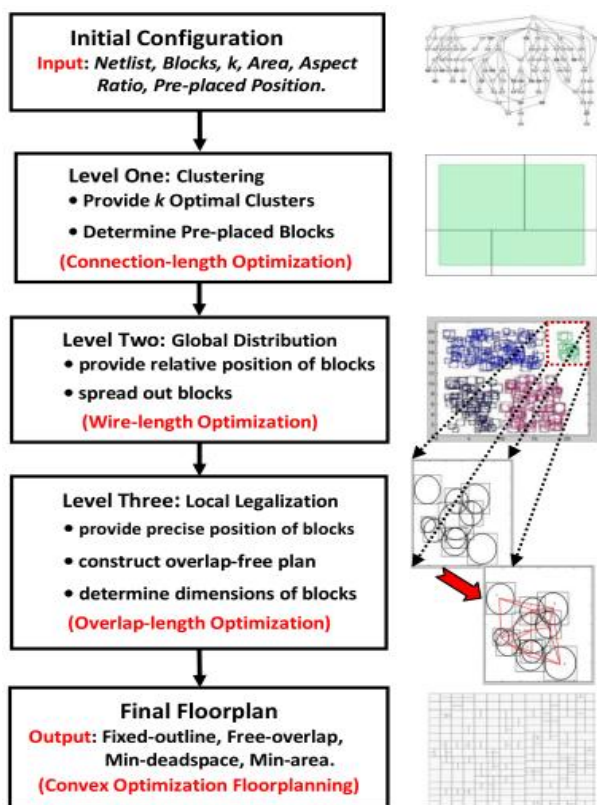


Figure 1. Proposed Multi-level Floorplanning Framework

A floorplanning approach decreases wire length to improve performance and reduce energy consumption. However, as power density becomes more severe, floorplanning techniques should consider fixed-outline design as well as other typical design metrics such as area and wire length. The importance of the fixed-outlining is shown in [7]. This is significantly difficult than outline-free floorplanning [7], [8] and typically aims to optimize the circuit performance: critical path delay; total circuit wire length; timing performance estimation; temperature dissipation; static power consumption; rout ability; and the multiple criteria.

In addition to the fixed-outline constraints, the placement regions often contains some pre-modules such as radio frequency [15]. These modules may contain pads inside its modules. Only when the locations of all pads in a chip are assigned, the bonding rules can be checked. Thus, we should place such kind of modules earlier than others. The other reason is that designers hope to place some modules closed to each other for timing or power consideration. Thus, the ability to handle pre-placed modules (PPM) is very important to a floorplanning algorithm [15].

A floorplan designing is a time consuming process. It is usually solved as a sequence of intractable sub problems consisting of initial configuration, relative module positions, and module positions precises, wire length minimization, aspect ratio satisfaction, complete area utilization and final floorplan optimization. In modern VLSI design, the size of the fixed-die is usually pre-determined during the chip synthesis process. Therefore, it is necessary to speed generate accurate none-overlap layout to satisfy area and timing constraints for a fixeddie packaging. This needs to reduce the time-to-market of optimal result generation to minimize

wire length and remove overlap modules on satisfying area and timing constraints [8]. Now, Fixed-outline against classical variable-die floorplanning must attempt to design on many constraints with a best result and reasonable time-to-market generation [7].

1.2. Proposed Framework

In the real design, there may exist several preplaced blocks in a die before floorplanning, which can be considered as a barrier. In order to get a feasible floorplan, it is necessary to keep away from these regions while placing blocks. Thus, we would like to handle fixed-die floorplanning with pre-placed blocks in these regions. Finding the optimal regions is very helpful to achieve the final floorplanning objectives. This is the main idea of our framework that divides the chip outline into several optimal regions (e.g. clusters). We find the optimal k clusters and preplaced blocks into these clusters to minimize our objectives in several levels. We propose a three-level fixed-outline floorplanning framework as a convex-optimization method for global optimization of wire length and chip area satisfaction. A flow chart of our proposed multi-level floorplanning framework (MFF) is given in figure 1 that includes three levels and several tasks. Note, there are three major frameworks as the flat, the hierarchical and the multi-level (hybrid hierarchical) to chip design [17]. Our MFF is a heuristic multi-pass SA method (with a rough ordering pass and a fine tuning pass) to present an accurate speed optimizer. This is an open mathematical domain research for optimizer design that helps to pick the best steps of solving an optimization problem [24].

At the first level of MFF, clustering methods try to classify the blocks based on its relations as a best proposed initialization of the planning (to minimize the connection-length). Our clustering algorithm is the best distribution of in itialized modules with speed convergence. In the second level, an attractor-repelled convex optimization model provides the relative positions of the modules on the floorplan (to minimize the wire-length). The third level places and sizes the modules using convex optimization (to minimize the overlap-length). Given the relative positions of the modules, an efficient method for generating relative position matrices with an interchange-free algorithm to local improvement of the floorplan is also presented. Since design computation is time-consuming and hard to formulate, we proposed a novel framework to design the levels of the blocks.

The experimental results of the standard benchmarks demonstrate an overlap-free and min-dead space floorplan that obtain significant improvements on the wire length and running time with complete chip area utilization. To the best of our knowledge, this is a speed multi-level convex design method that used for fixed-outline floorplanning. The proposed MFF is particularly suitable for fixed-outline that can be used to classical floorplanning.

1.3. Paper Motivations

We pointed out six main problems with floorplanning: (1) a lack of attention to the fixed-outline constraints; (2) a lack of emphasis on convex optimization floorplanning and corresponding benchmarks; (3) a lack of attention to multi-level objective optimization with discrete objectives; (4) a

Lack of consideration to outcome of initial module distribution as a clustering solution; (5) a Lack of attention to running time; (6) and an inability to handle scalability [8]. Some works proposed of fixed-outline floorplanning process as our motivation examples [3], [2] and [6]. They defined before research on fixed-die based by mathematical approaches without any respect to clustering that has been extensively studied in the image processing, data mining, and machine learning. Also, the convex modeling is a slow method and ignores to generate a speed framework for time-to-market reducing.

1.4. Paper Objectives

The above considered works have attempted to deal with some of the above issues. However, all concerns are difficult and needed, we cover the all points and focus on solving last four problems. We aimed to propose a speed three level framework based on [2] and [3] as a multi-level objective satisfaction method. The different of our framework are on outcome of initial modules distribution (as a clustering method) with a novel three level SA-based convex framework (see section 3) and better results on wire length, dead space and area utilization with valuable reducing the running time without any modules overlapping.

The focus of this paper is a novel three level floorplanning framework that implemented in a VLSI system with a new heuristic multi-pass SA method able to cope fixed-outline designing as well as variable conditions. The paper proposes a multi-Level framework using convex optimization. The aim of our work is to present techniques that use convex optimization to find global solution of an optimized floorplanner. Also, the runtime of the framework must be reasonable to a typical mathematical solution. The optimal floorplanner minimizes the wire length by mathematical programming and reduces running time for effectiveness and efficiency, scalability and robustness, stability, flexibility, optimality and applicability of the result. This can minimize the area of variable-outline to cover classical floorplanning.

1.5. Paper Contributions

In the real design, there may exist several preplaced blocks on a die before floorplanning, which can be considered as a barrier. In order to get a feasible floorplan, it is necessary to keep away from these regions while placing blocks. Thus, we would like to handle fixed-die floorplanning with preplaced blocks in these regions. Finding the optimal regions is very helpful to achieve the final floorplanning objectives. This is the main contribution of our proposed framework that we divide the chip Outline into several optimal regions (e.g. cluster). We find the optimal k clusters and pre-place blocks into these clusters to minimize our objectives in several levels.

However, this is the first time that a convex optimization based method has been used for fixed-outline floorplanning and some contributions are proposed in our work that is comparable with different recent researches as explained in paper motivations (see Section 1.3). Our contributions in this framework are divided into three main fold. First, we implement our framework on a new multi-pass SA method.

Second, we propose an efficient objective clustering algorithm to place the chip die blocks. We show this idea is an effective way for reducing wire length in addition to changing block's position on a floorplan with speed convergence. Third, we present a two-stage convex optimization-based technique for fixed-outline floorplanning to minimize wire length and area satisfaction concurrently.

Since accurately solving wire equations is a time consuming task with high complexity, we propose a scalable framework to find convex solution of the floorplan for faster generation a global optimal result. Based on presenting an initial configuration in VLSI design as a new level with optimal clustering of the blocks, we present a multi-pass SA-based method which focuses on a speed mathematical implementation of our problem solving. Our key contributions are summarized as below:

1. **Framework:** A practical heuristic multi-pass SA-based method to catch speed and accurate global optimization with a new clustering level.
2. **Model:** A convex-optimization based modeling to solve an NP problem with a reasonable running time.
3. **Scope:** A scalable framework under fixed-die constraints to complete area utilization with greater improvement on wire length, dead space, area and running time.
4. **Capability:** A framework enables to variable-die and PPM floorplanning.

The first contribution is not mentioned in early works and need to apply on mathematical solving implementation for initial point selections; the second is proposed in earlier works without any respect to speed result generation; the third does not apply to before frameworks and needs for future massively floorplanning; the last contribution shows the floorplanner can apply to classical methods.

1.6. Paper Organization

The remainder of this paper is organized as follows. In Section 2 related works are presented. Section 3 describes the preliminaries and the problem definition. The proposed framework is explained in Section 4. Experimental results are provided and discussed in Section 5. Finally, Section 6 concludes the paper and mentions some future directions.

2. Related Work

Some of floorplanners for solving the NP-hardness of its floorplanning are employed heuristic approach [10]. However, there are a number of heuristic methods that used in floorplanning e.g. Simulating Annealing (SA) [33], Genetic Algorithms (GA) [32] or Evolutionary Algorithms (EA) [30], SA is preferred. The SA-based method is preferable in flat framework floorplanners because of its simplicity and applicability. SA is a local search based heuristic approach and SA based floorplanners tend to be trapped in local optimal solution. Therefore, SA needs to move on convex optimization modeling. Convex modeling of floorplanning is new and an important property of the method is that any local result is a global for a given problem [11]. For related work investigation, we focus on some works that used the convex programming and SA models on fixedoutline to present an accurate and practical method.

There have been presented a few research on fixed-outline methods that use the convex modeling. Floorplanning problem formulated as a geometric program based on convex optimization modeling in [2]. The same very time-consuming method presented in [4] and combined the SA with convex optimization to find zero-dead space results. They reduced the number of variables and functions to improve the efficiency of algorithm with long time to find good solution. For future nanometer scale hierarchical design, these classical methods cannot satisfy the needed requirements. Also, in modern ASIC design to fixed-outline floorplanning we prefer to select a hierarchical framework [8]. Next, an annealing-based fixed-outline floorplanning method proposed a slack-based algorithm to shape the soft blocks [7]. Such methods with shaping algorithm are two main problems. There is no optimality guarantee to resulting solution and the fixed-outline constraint produced a penalty in annealing cost function. For solving these, a method determined the block shape using second-order cone programming [2], the aspect ratio constraint on each block was ignored. So, in non-slicing floorplan it is necessary to efficient and optimal design and shaping algorithm. This is formulated for fixed-outline to consider the aspect ratio constraints.

An analytical approach proposed a fixed-outline floorplanning with soft modules to minimize wire length in two stages [22]. In the first stage, the wire length and distribution density of modules are minimized. In the last stage, the overlap problem and wire length are optimized to obtain an overlap-free floorplan. Recently, a convex optimization method presented with two stages [3]. To minimize the total wire length in first, relative positions of modules used a convex optimization model, named attractor-repelled (AR). To minimize wire length and facilitates dead space-free with overlap-free floorplans, the second stage determined the actual locations and shapes of blocks. The used AR model cannot uniformly distribute blocks to the placement region. So, after the first stage the relative positions are completely different and needed to consider wire length again.

Also, a fixed-outline method is described to solve a group of four quadratic equations iteratively to facilitate its integration with SA, called SAFFOA [6]. Another multilevel two-phase floorplanner developed a top-down partitioning and a bottom-up merging to optimize total wire length with min-cut partitioning via SA-based method [17].

However, these methods are well suited and applicable to floorplanning process; they are time consuming approaches without any stability on finding optimal solution by convex modeling, scalability on floorplan size and a reasonable runtime. Our motivated papers are presented as fixed-outline with convex modeling on wire-length minimizing ([3], [2] and [6]). However, there are recent contributed methods on fixed-die that moved on convex modeling, we aimed to reduce the running time and improvement on wire length minimizing with complete area utilizing to find a scalable approach. We apply mathematical convex programming on three level multi pass SA-based method to cope the local optimal trap. Our results show a scalable framework with significant improvement on optimal design of wire length and reduction of the running time comparable to the before important works ([2], [6], [3], [16], [15]).

Although our method looks a bit similar to our motivational referenced methods ([2], [6], [3], [16], [15]), they are different in many ways. In the first, our reference models do not focus on clustering as a new level and randomly select the initial blocks condition. The second, we use the actual length of two connections while [3] uses quadratic length; we compute wire length by half-perimeter wire length (HPWL) that is more accurate than [3]. The third, while [3] needs to formulate and minimize wire length in stage two, we do not model the wire length in our third level; thus, our objective is only to meet the outline constraint. We compute well wire length in the second level and remove this objective from the third level objectives. Also, they record all geometric relations between each pair of modules and, we record the necessary geometric modules relations.

3. Problem Definition

A formal presentation of the floorplanning problem with definition and formulation on inputs, outputs and objective function (or cost function) is described as follows.

3.1. Floorplanning Inputs

The inputs of a for planning process are defined as below (see figure 1):

- A list of areas $A_i, 1 \leq i \leq n$ with a set of n rectangular modules $M = \{m_1, m_2, \dots, m_n\}$;
- a partition of M is defined as set of modules with fixed orientations into set M_1 and is defined with free orientations into set M_2 ;
- the connectivity between modules is represented by an interconnection matrix $C_{n \times n} = [c_{ij}], 1 \leq i, j \leq n$ by a netlist, that define the connectivity between two modules i and j as c_{ij} with assuming C is symmetric, i.e., $c_{ij} = c_{ji}$;
- the area of each module i is represented by A_i ;
- width, height, and area of each i , are denoted by w_i , h_i , and A_i , respectively;
- the aspect ratio R_i of each module i is bounded to R_i^{low} and R_i^{up} .
- the acceptable aspect ratios are in the range $[L_i, U_i]$;
- an outline-free floorplanning instance is bounded by w_F^{low} and w_F^{up} for width, and h_F^{low} and h_F^{up} for height of the floorplan;
- given a die, its dimension is fixed and the width and height are denoted by W_f and H_f , respectively;
- and a fixed-outline floorplanning instance is presented by area value A of the floorplan.

3.2. Floorplanning Outputs

The outputs of a Floorplanning process are described as below (see figure 1):

- A floorplanning problem is defined by the location, width, and height of each module on a floorplan instance;
- all the modules are enveloped in the floorplan;
- there is not any overlapping between the modules $R_i \cap R_j = \emptyset$ for $i \neq j$;
- module i with $h_i \times w_i = A_i, 1 \leq i \leq n$ is defined by (x_i, y_i) as coordination, w_i as width and h_i as height;

- module i with fixed orientation ($i \in M_1$) is defined by $R_i^{low} \leq \frac{h_i}{w_i} \leq R_i^{up}$;
- module i with free orientation ($i \in M_2$) is defined by $R_i^{low} \leq \frac{h_i}{w_i} \leq R_i^{up}$ or $\frac{1}{R_i^{up}} \leq \frac{h_i}{w_i} \leq \frac{1}{R_i^{low}}$;
- and the objective function is formulated as an optimization problem

3.3. Floorplanning Objective Function

An optimal floorplan is defined by the desired objective function [30]. The floorplanning is defined as given n modules with the fixed chip area A of the floorplan, the objective is to place all the modules inside the area without any overlapping, minimize the total wire length with area satisfaction such $\sum_{i=1}^n A_i = A = W_f \times H_f$ where W_f and H_f are the fixed width and height of the floorplan. In our framework, the sum of each area of the modules is equal to the area A of the chip. We enforce this constraint into the second and third levels formulation by the height H_f and width W_f of the chip. ZDS is now regarded as a constraint in the modeling and we are capable of obtaining ZDS solution and also allowing any percentage of dead space in the fixed-outline floorplan. The objective function of a floorplanning process is formulated as (1) [31]. Our objectives minimization for the first, the second and the third levels are connection-length, wire-length and overlap-length, respectively.

$$\min \sum_{i=1}^l w_i \times \left(\frac{f_i - f_i^*}{f_i^*} \right) \quad (1)$$

where l is the number of objectives (in our model l is equal to 1 for each level); f_i is objective function i ; f_i^* is an optimized value to objective function i that find for a single objective function f_i ; w_i is the weight of objective i ; our objectives are connection-length, wire-length and overlap-length. If the objective functions are not all in the same scale, $\left(\frac{f_i - f_i^*}{f_i^*} \right)$ is formulated to normalize them. All weights are used as $(\sum_{i=1}^l w_i = 1)$. Note that our objectives are satisfied with different functions on different three levels of the proposed framework to reduce the problem complexity.

4. Proposed Framework

A generic placement flow with three steps is used by most floorplanners today as global placement, local legalization and detailed placement. Global placement with a global view processes all blocks together and defining a location for each of them. Blocks are distributed across the die surface with a quite larger execution time that is comparable to the other sections. So, for reducing the total execution time of a scalable floorplanner, we need to try further optimization during this step. Our first motivated and contributed is added clustering level to reduce the problem size. Our second motivated is convex modeling and accurate wire length calculation to increase the optimized result accuracy. After completion of the global planning may exist overlap area among the blocks and poses a lot of difficulties. So, we do the overlapped area removing during the legalization step. Finally, for optimizing the solution plan, we may need some local changes to improve the objective function.

Since, solving related mathematical equations accurately is a time consuming method, the main solved bottleneck of our MFF is computing needed objectives concurrently. Another problem is finding accurate results of the floorplanning as a global optimization model with reasonable running time. These are very difficult and time consuming on fixed-die and min-dead space problems. Inconvenient initialization of the problem formulated and same objective satisfaction on different locations of the solving model can change it to an unsolved non-convergence problem.

Based on the above description, in this paper, we propose a framework (see figure 1) with different objectives on each level to reduce the problem complexity. This is implemented by a multi-pass SA-based method to speed convergence and joined with convex model to produce an accurate optimized result. This is a global optimization problem for wire length minimization and total area utilization. Good problem initialization with optimal clustering can help us to speed solving the problem with rapid convergence on a multi-level method. Next two levels of MFF are the global distribution to obtain minimum wire length of the fixed-die and the legalization to obtain feasible solution. At the first level, we applied a novel clustering method for initial distribution of the modules. This is provided the best initialize relative positions of the soft modules to the next level to reduce the running time of our framework.

Algorithm 1 Multi-Pass SA-based Optimization Method of MFF

```

1: Procedure MSOM( $R, r, Kmax, F, S$ ) {Take two constant step size variables ( $R$  and  $r$ ),
total algorithm reputation ( $Kmax$ ), number of iterations ( $F$ ), a start point ( $S$ )
and finally return an answer with a best start point ( $S$ )}
2: Compute  $f = \frac{Kmax}{F}$ ; {Compute the number of repetitions ( $f$ ) for each iteration ( $F$ )}
3: SACostFunction( $S$ ); {Compute SA Cost Function with initial start point  $S$ }
4: for  $i \leftarrow 1$  to  $\frac{f}{2}$  do
5:   {Running SA with Large Step}
6:   for  $j \leftarrow 1$  to  $f$  do
7:     SA( $R, S$ ); {Run SA with start point  $S$ }
8:     SACostFunction( $S$ ); {Compute SA Cost Function of new start point}
9:     if ( $SACostFunction(S) \geq SACostFunction(S^*)$ ) then
10:       $S \leftarrow S^*$ ; {Selects a better start point as  $S^*$ }
11:   {Running SA with Small Step}
12:   for  $k \leftarrow 1$  to  $f$  do
13:     SA( $r, S$ ); {Run SA with start Point  $S$ }
14:     SACostFunction( $S$ ); {Compute SA Cost Function of New Start Point}
15:     if ( $SACostFunction(S) \geq SACostFunction(S^*)$ ) then
16:       $S \leftarrow S^*$ ; {Selects a better start point as  $S^*$ }
17: return ( $S$ );
```

Algorithm 2 Proposed Clustering Algorithm

```

1: Procedure NCut ( $W, k$ )
   {Take a similarity matrix (weight  $W$ ) and cut the graph into  $K$  clusters}
2: Construct diagonal matrix  $D$ ;
3:  $\hat{L} \leftarrow D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ ; {Compute the normalized Laplacian  $L$ }
4: Compute top  $k$  eigenvectors  $v_1, \dots, v_k$  of  $L$ ;
5: Form matrix  $k$  be the matrix containing the vectors  $v_1, \dots, v_k$  as columns;
6: for  $i \leftarrow 1$  to  $n$  do
7:    $y_{ij} = V_{ij} / \|V_i\|$ ; {Let  $y_i \in \mathbb{R}^k$  be the normalized vector corresponding
to the  $i^{th}$  row of  $V$ ; dimensionality reduction:  $n \times n \rightarrow n \times k$ }
8: Cluster the points  $(y_i)_{i=1, \dots, n}$  in  $\mathbb{R}^k$  with the  $k$ -means algorithm into clusters  $C_1, \dots, C_k$ ;
9: return Clusters  $A_1, \dots, A_k$  with  $A_i = \{j | y_j \in C_i\}$ ;
```

This is designed to decrease the total running time of our MFF with connection-length minimization. In the second level, we made a convex model to the original fixed-die floorplanning problem. This global optimization problem provided the relative positions of the soft modules on the floorplan while the total wire length was minimized based on AR model [3]. In the third level, a fixed-die floorplanning problem was formulated to a nonlinear optimization problem.

The relative positions of all modules encode to avoid the operations of absolute values in the non-overlap constraints.

Our heuristic MFF is a multi-objective convex model that find its objectives on the three levels, separately (see figure 1) [30]. Each level has its own objective function to reduce the complexity and the running time of the proposed MFF. A problem size reduction technique is defined as the process by which large problems are reduced into relatively smaller problems that can be easily solved by using our SA-based solver. In remaining of this section, we first describe our multi-pass SA-based method and three different levels of our MFF as clustering, global distribution and local legalization of each predefined clusters.

4.1. Multi-Pass SA-Based Method

Our recommended Multi-Pass usage of the MFF where two passes are performed on the same underlying model is described as Algorithm 1. The first pass is a rough ordering pass with big step and small running time. The second pass is the fine tuning pass that has a small initial step and big running time. At first, for each iteration selects the number of repetitions. Next, runs the SA in two passes with different steps. Finally, chooses the best result of the problem with selecting a well interior point.

4.2. Level One: Clustering

Clustering is a task of grouping a block set in such a way that block in the same cluster are more similar to each other than to those in other clusters [26]. Once all the blocks have been grouped together into clusters, each of these clusters can be represented like a virtual block and treated similar to regular blocks. This can be reduced the problem into smaller chunk and solving them independently. For very large designs the floorplan problem gets reduced to the placement of a few hundreds of virtual blocks instead of 1 million standard blocks, thus reduces the complexity of our processing. This affects the quality of the final solution that is better routable and scaled up for larger designs [21].

Spectral clustering techniques make use of the eigenvalues of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions. It refers to a set of heuristic algorithms, all based on the overall idea of computing the first few singular vectors and then clustering in a low dimensional subspace [27]. The aim of this level is reducing the area (bus area). The objective function of this level is turned into an eigenvalue problem, which can be solved deterministically in polynomial time. This is a major advantage over constrained k-means clustering, which produces non-deterministic solutions.

We focus on using blocks clustering as an approximation solution for k-way graph partitioning problem (Algorithm 2) and try to solve the problem using normalized-cuts clustering techniques [27, 26]. Initial Block Clustering (IBC) algorithm uses a Laplacian matrix L and normalizes the eigenvectors before using k-means to cluster them into k partitions. Most stable clustering methods are usually given by the value of k that maximizes the eigengap (difference between consecutive eigenvalues). All needed definition for the algorithm is described as follows.

For an undirected weighted graph $G = (V, E)$ in the clustering algorithm, the vertices in G is a set of vertices needs to be clustered into k clusters. Various ways can be used to compute the edge weight between each pair of vertices. These weights form the weight matrix W , where $w_{ij} = w_{ji} = 0$. A vertex degree $v_i \in V$ is defined as $d_i = \sum_{j=1}^n w_{ij}$. We define the degree matrix D as the diagonal matrix with the degrees d_1, d_2, \dots, d_n on the diagonal. Two normalized graph laplacian matrix (L_{sym} and L_{rw}) are defined as (2) and (3) with adjacency matrix W . Important notations used throughout this section are listed in table 1.

$$L_{sym} = D^{-1/2} L D^{-1/2} = 1 - D^{-1/2} W D^{-1/2} \quad (2)$$

$$L_{rw} = D^{-1} L = 1 - D^{-1} W \quad (3)$$

We construct a partition by solving the mincut problem that consists of choosing the partition A_1, A_2, \dots, A_k k and minimizes (4).

$$\text{cut}(A_1, A_2, \dots, A_k) = \sum_{i=1}^k \text{cut}(A_i, \hat{A}_i) \quad (4)$$

To cope imbalance graph partitioning by mincut, the objective function needs to guarantee that sets A_1, A_2, \dots, A_k can be reasonably large. The encode objective function is the normalized cut $Ncut$ as (5) while the size of a subset A of a graph is measured by the weights of its edges $\text{vol}(A)$. Finding the optimal $Ncut$ is NP hard; the above clustering algorithm is a relaxation of the Normalized Cut algorithm.

$$Ncut(A_1, A_2, \dots, A_k) = \sum_{i=1}^k \frac{\text{cut}(A_i, \hat{A}_i)}{\text{vol}(A_i)} \quad (5)$$

Table 1. The Clustering symbol table

Symbol	Definition
G	A (weighted) graph
A	The affinity matrix
D	The degree matrix
I	The identity matrix
L / \hat{L}	The unnormalized/normalized graph Laplacian
Q / \hat{Q}	The unnormalized/normalized constraint matrix
vol	The volume of graph G

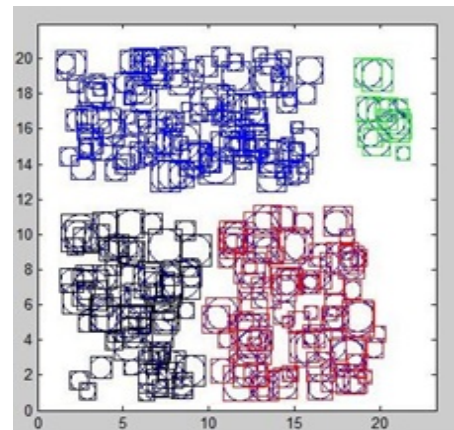


Figure 2. Clustering and Distribution Result of n300

In this section, we focus on using blocks clustering to try to solve graph partitioning problem for large scale floorplanning and finding an optimal value for k (see figure 2). With the obtained partitioning value, our framework is

initialized and continued. We try to produce optimal k clusters to present an initial blocks configuration to next levels. If we applied k-means to laplacian eigenvectors allows us to find cluster with non convex boundaries.

4.3. Level Two: Global Distribution

In the global distribution level with a convex optimization model, we uniformly spread modules among all k clusters and concurrently minimize the total wire length. It is very difficult to balance these two conflicting objectives. Our model uses two forces, which are attractive force and repulsive force, to compromise these two factors. If two modules have large numbers of interconnections, it would pull two modules to closer locations to minimize wire length. On the contrary, when there exist a serious overlap of any two modules, it would add an additional penalty such as two modules are pushed away from each other. By properly formulating these two forces, the minimal value can happen when the two circles touch at one point. That is why we can uniformly distribute circles that represent modules among the placement clusters and simultaneously minimize wire length.

At this level, may be exist some overlaps between the modules. Therefore, we need to remove the module overlapping, which is considered at next third level. Since modules have been uniformly distributed among the region with small wire length, we only need to do non-overlapping modules without considering the wire length further more. Also, we extract geometric relations of modules from their distribution to find good result.

We record them by a horizontal and a vertical constraint graph. To maintain the necessary relations, the constraint graphs are built according to the triangles obtained by applying the Delaunay triangulation (DT) method [28] to the distribution. Then, the other convex optimization function as well as a lot of constraints transformed from the graphs is used to meet the desired objective in the model, each block m_i is transformed into a circle C_i . Its coordinate is denoted by (X_i, Y_i) , and the radius is denoted by r_i , which is proportional to $\sqrt{A_i}$. The connectivity between blocks m_i and m_j , and d_{ij} denoted by c_{ij} and is the square distance between the centers of the corresponding circles as (7). The necessary equations are proposed as follows:

$$s_{ij} = (r_i \times r_j)^2 \quad (6)$$

$$d_{ij} = \sqrt{((X_i - X_j)^2 + (Y_i - Y_j)^2)} \quad (7)$$

$$p_{ij} = (r_i + r_j) - d_{ij} \quad (8)$$

We aim to plan a given set of soft blocks inside a fixed-outline floorplan with min-dead space and/or free overlap among the blocks. Our model uses d_{ij} to measure accurate wire length with convex approximation of the HPWL [14]. The model wants to give a larger force to push circles away if they overlap. However, once they are separate, the force should be reduced gradually. The minimum value with two circles just touched at one point is $(d_{ij} = (r_i + r_j))$ (i.e., $p_{ij} = 0$ in (8)). The model of two circles C_i and C_j contains two convex functions for the circles overlapping ($p_{ij} = 0$) and non-overlapping is ($p_{ij} < 0$) as:

$$f_{ij}(X_i, X_j, Y_i, Y_j) = \begin{cases} c_{ij}d_{ij} + s_{ij} \frac{p_{ij}}{d_{ij}}, & p_{ij} \geq 0 \\ c_{ij}d_{ij} + 1 \times \frac{p_{ij}}{d_{ij}}, & p_{ij} < 0 \end{cases}$$

We have two states for two circles as overlap (e.g. $p_{ij} \geq 0$) and non-overlap (e.g. $p_{ij} < 0$). We model the overlap state as (9) and its first derivation as (11). The long and short distances of two circles are the answer of non-overlap and overlap, respectively. Our model is a convex function and the minimum value happens for $(r_i + r_j)$.

$$f(d_{ij}) = c_{ij}d_{ij} + s_{ij} \frac{((r_i+r_j)-d_{ij})}{d_{ij}} \quad (9)$$

$$= c_{ij}d_{ij} + s_{ij}(r_i + r_j)d_{ij}^{-1} - s_{ij} \quad (10)$$

$$f'(d_{ij}) = c_{ij} - s_{ij}(r_i + r_j)d_{ij}^{-2} = 0 \quad (11)$$

$$d_{ij} = \pm \sqrt{\frac{s_{ij}(r_i+r_j)}{c_{ij}}} \quad (12)$$

The objective function of this level is defined as (13) and its constraints guarantee all blocks placed in the die.

$$\min_{(X_i, Y_i, X_j, Y_j)} \sum_{1 \leq i < j \leq N} f_{ij}(X_i, Y_i, X_j, Y_j) \quad (13)$$

Subject to

$$\begin{aligned} X_i + r_i &\leq W_f, \\ X_i - r_i &\geq 0, \\ Y_i + r_i &\leq H_f, \\ Y_i - r_i &\geq 0. \end{aligned}$$

where H_f, W_f is the height and width of the floorplan; N is the number of circles.

Herein, we show that the objective function of this level is a convex function. To present convexity property of the function, we use the well-known fact that is stated if $f: \mathcal{R}^n \mapsto \mathcal{R}$ is twice continuously differentiable, then f is convex on an open convex set $C \subset \mathcal{R}^n$ if $\nabla^2 f_{ij}$ is positive semidefinite for all points in C [31]. Namely, it is adequate to prove the Hessian is positive semidefinite. It is easy to show that $\nabla^2 f_{ij}(x_i, x_j, y_i, y_j) = 0$ [2]. So, the Hessian of f_{ij} is positive semidefinite. Since f_{ij} has a four variables x_i, x_j, y_i and y_j the Hessian of f_{ij} is a 4×4 matrix, which is the second derivation of f_{ij} . Thus, the Hessian matrix of f_{ij} is as follows:

$$\begin{aligned} H(f_{ij}(X_i, Y_i, X_j, Y_j)) &= \nabla^2 f_{ij}(x_i, x_j, y_i, y_j) \\ &= \begin{pmatrix} \frac{\partial f_{ij}}{\partial x_i^2} & \frac{\partial f_{ij}}{\partial x_i x_j} & \frac{\partial f_{ij}}{\partial x_i y_i} & \frac{\partial f_{ij}}{\partial x_i y_j} \\ \frac{\partial f_{ij}}{\partial x_i x_j} & \frac{\partial f_{ij}}{\partial x_j^2} & \frac{\partial f_{ij}}{\partial x_j y_i} & \frac{\partial f_{ij}}{\partial x_j y_j} \\ \frac{\partial f_{ij}}{\partial x_i y_i} & \frac{\partial f_{ij}}{\partial x_j y_i} & \frac{\partial f_{ij}}{\partial y_i^2} & \frac{\partial f_{ij}}{\partial y_i y_j} \\ \frac{\partial f_{ij}}{\partial x_i y_j} & \frac{\partial f_{ij}}{\partial x_j y_j} & \frac{\partial f_{ij}}{\partial y_i y_j} & \frac{\partial f_{ij}}{\partial y_j^2} \end{pmatrix} \\ &= \begin{pmatrix} \tau & -\tau & \upsilon & -\upsilon \\ -\tau & \tau & -\upsilon & \upsilon \\ \upsilon & -\upsilon & \varphi & -\varphi \\ -\upsilon & \upsilon & -\varphi & \varphi \end{pmatrix} \end{aligned}$$

where τ , φ and ν are as follows:

$$\begin{aligned}\tau &= c_{ij}d_{ij}^{-1} - (c_{ij}(X_i - X_j)^2 + s_{ij}b) d_{ij}^{-3} \\ &\quad + 3s_{ij}b(X_i - X_j)^2 d_{ij}^{-5} \\ \varphi &= c_{ij}d_{ij}^{-1} - (c_{ij}(Y_i - Y_j)^2 + s_{ij}b) d_{ij}^{-3} \\ &\quad + 3s_{ij}b(Y_i - Y_j)^2 d_{ij}^{-5} \\ \nu &= (X_i - X_j)(Y_i - Y_j)(-c_{ij}d_{ij}^{-3} + 3s_{ij}bd_{ij}^{-5})\end{aligned}$$

4.4. Level Three: Local Legalization

In global floorplan techniques using clustering, legalization is necessary which is required to eliminate the overlaps caused by restricting the boundaries of the clusters. At second level, the circles that represent blocks have been uniformly distributed over a specified region and the wire length is minimized. Now, we have to determine the exact locations and shapes of blocks to remove any overlapping and arrange all the blocks inside the die. To maintain the optimized wire length of before level, we extract geometric relations of blocks from the circles distribution and record the necessary relations by constraint graphs. A good wire length obtained by before level and we do not consider wire length again; our objective only intends to satisfy the outline constraints.

To obtain overlap-free floorplans, we use Second Order Cone Programming (SOCP) formulation [34]. In other word, given the fixed-outline of the floorplan and the locations of circles (from the second level), semidefinite programming is used to provide the precise location and rectangular dimensions of the blocks. Convex programming SOCP is used to formulate a minimization linear objective function subject to SOC constraints. So, the third level model is formulated by applying semidefinite optimization techniques to the area and aspect ratio constraints using the relative position of the blocks (e.g. locations of circles).

We use relative position matrices (RPM) to encode the relative positions of the blocks that can avoid the operations of absolute values in the non-overlap constraints. We use sparse relative position matrixes (SRPM) further to improve computational efficiency. We use DT technique to spread out the blocks in the plan obtained from the second level and convert it into a planar graph to produce the voronoi diagram [28]. The neighboring objects are identified by an edge in the resulting DT graph as vertical and horizontal relations. An interchange-free algorithm for local improvement of the plan is used that achieves desired aspect ratio constraints on the soft blocks.

To satisfy the Fixed-outline constraint, the objective function tries to minimize the deference between predefined height (width) and the resulting height (width) according to the objective function. By defining $|W - W_f|$ and $|H - H_f|$, we formulate the objective function for fixed-outline floorplanning with non-overlap constraint as (14).

$$\min \{|W - W_f| + |H - H_f|\} \quad (14)$$

subject to

$$\begin{aligned}x_i + w_i &\leq x_j, \forall (n_i, n_j) \in C_h \\ y_i + h_i &\leq y_j, \forall (n_i, n_j) \in C_v \\ \forall i, x_i + \frac{1}{2}w_i &\leq \frac{1}{2}w_F,\end{aligned}$$

$$\begin{aligned}\forall i, y_i + \frac{1}{2}h_i &\leq \frac{1}{2}h_F, \\ \forall i, \frac{1}{2}w_i - x_i &\leq \frac{1}{2}w_F, \\ \forall i, \frac{1}{2}h_i - y_i &\leq \frac{1}{2}h_F, \\ \forall i, w_F^{low} &\leq w_i \leq w_F^{up}, \\ \forall i, h_F^{low} &\leq h_i \leq h_F^{up}, \\ \forall i, \left\| \begin{pmatrix} w_i - h_i \\ 2\sqrt{A_i} \end{pmatrix} \right\|_2 &\leq h_i + w_i, \\ \forall i, \left\| \begin{pmatrix} A_i - \beta_i^* \\ 2h_i \end{pmatrix} \right\|_2 &\leq A_i + \beta_i^*, \\ \forall i, \left\| \begin{pmatrix} A_i - \beta_i^* \\ 2w_i \end{pmatrix} \right\|_2 &\leq A_i + \beta_i^*,\end{aligned}$$

where C_h and C_v are vertical constraint graph and horizontal constraint graph, respectively; $\|\dots\|_2 \leq \dots$ is the standard SOC constraint. We relax the area and the aspect ratio constraints and formulate to SOC constraints. The area constraint, $w_i \cdot h_i = A_i$, of each soft block is relaxed as $w_i \cdot h_i \geq A_i$. For $A_i \geq 0$, this constraint formulates to SOC as (15) and inserted to the constraints of (14). This process calculates as follows:

$$\begin{aligned}w_i \cdot h_i &\geq A_i \\ h_i^2 + 2w_i \cdot h_i + w_i^2 &\geq (h_i^2 - 2w_i \cdot h_i + w_i^2) + 4A_i \\ (w_i + h_i)^2 &\geq (w_i - h_i)^2 + 4A_i \geq 0\end{aligned}$$

$$\begin{aligned}(w_i + h_i)^2 &\geq \sqrt{(w_i - h_i)^2 + (2\sqrt{A_i})^2} \\ \forall i, (w_i + h_i) &\geq \left\| \begin{pmatrix} w_i - h_i \\ 2\sqrt{A_i} \end{pmatrix} \right\|_2\end{aligned} \quad (15)$$

Also, the aspect ratio B_i of block i is formulated as $B_i = \frac{\max\{w_i, h_i\}}{\min\{w_i, h_i\}}$. We assume that the aspect ratio of block i must be bounded by a given value $B_i^* > 0$. If $w_i^{low} = h_i^{low} = \sqrt{A_i/\beta_i^*}$ then $w_i \geq w_i^{low} > 0$, $w_i^2 \geq A_i/\beta_i^*$, $\beta_i^* w_i^2 \geq A_i$, $\beta_i^* \geq w_i/h_i$. Also, for $h_i \geq h_i^{low} > 0$ we obtain $\beta_i^* \geq h_i/w_i$. With $\beta_i^* \geq h_i/w_i$ and $w_i \cdot h_i = A_i$, we obtain $\beta_i^* \geq h_i^2/A_i$. So we obtain $\beta_i^* A_i \geq h_i^2$. Combining $\beta_i^* \geq w_i/h_i$ and $w_i \cdot h_i = A_i$ results $\beta_i^* \geq w_i^2/A_i$. There for, $\beta_i^* A_i \geq w_i^2$. So, the aspect ratio constrains for height and width of each block formulates same as (15) that we show in the constraints of (14).

Note that the objective function of the model is the rectilinear distance between blocks b_i and b_j (e.g. $|x_i - x_j| + |y_i - y_j|$). Also, there are not non-overlap constraints in the constraints of (14). They are formulated to non-linear equations (e.g. (16) and (17)) and must be converted to linear as (18) and (19). This describes as follows.

$$\frac{1}{2}(w_i + w_j) \leq |x_i - x_j|, \text{ if } |y_i - y_j| < \frac{1}{2}(h_i + h_j) \quad (16)$$

$$\frac{1}{2}(h_i + h_j) \leq |y_i - y_j|, \text{ if } |x_i - x_j| < \frac{1}{2}(w_i + w_j) \quad (17)$$

By using the relative positions of the blocks from the second level, we linear zed the (16) and (17) equations. For this, we determine which of the two conditions must be applied by vertical and horizontal relative positioning and eliminate the absolute value function. To do the above objective, we define $\Delta y \geq \Delta x$ for the vertical relative positioning and $\Delta x > \Delta y$ for the horizontal relative

positioning. To model the vertical relative positioning, if $y_i \geq y_j$, we apply the Equation (18) to the model and if $y_j \geq y_i$, we substitute the Equation (19). We did the same action to define the horizontal relative positioning.

$$\frac{1}{2}(h_i + h_j) \leq y_j - y_i \quad (18)$$

$$\frac{1}{2}(h_i + h_j) \leq y_i - y_j \quad (19)$$

5. Experimental Results

Our framework is experimented using Microelectronics Center of North Carolina (MCNC) and Gig scale Systems Research Center (GSRC) standard benchmarks. The experiments demonstrate the efficient enhancements on wiring, run time and area results compared to the based approaches. In this section, we proposed and evaluate the framework efficiency by many experiments. At first we explain the experimental setup, methods for comparison, and then continue with results evaluation. Especially, our clustering applied model is analyzed by scalability and complexity analysis. The running time of our applied methods and SA convergence is explained.

5.1. Experimental Setup

Our proposed framework to evaluate its efficiency has been used the GSRC [18] and the MCNC [19] benchmark suites. All needed benchmark suits properties are shown in table 2. We made simplicity on GSRC and MCNC in our experiments that all modules are soft blocks with fix area and variable dimension. We compute actual dimensions of blocks by using the area and the aspect ratio of the blocks. To adapt GSRC and MCNC for our experiments we assume pins are placed in the middle of block i with R_i^{up} and $R_i^{low} = 1/3$. Aspect ratio is the ratio of height/width and width/height in the blocks.

In our used version, the dimensions of the GSRC benchmarks are in the range of tenths of a Micron, and for the MCNC benchmarks the dimensions are in the range of microns thus it is necessary to scale them down to be applicable for 95nm technology. We estimated the wire length by using HPWL in all experiments.

We implemented our SA-based MFF and other algorithms in the MATLAB with our motivated works. Before each algorithm, we first use a simple python script to prepare the graph data into our needed matrix format. In the clustering algorithm the Laplacian matrix is symmetric positive definite. S Thus we use MATLAB function `eigs()` to compute the top k eigenvectors and for k -means step we apply MATLAB function `kmeans()`. The machine we use for this algorithm is an 1.7GHz Intel(R) Core(TM) i7 CPU with 8.0GB RAM and runs under MATLAB 8.2 (R2013b) in Windows 7.

5.2. Methods for Comparisons

We are going to compare the results of our framework with important methods (UFO [2], SAFFOA [6], LUO [3], TIM [16] and MK [15]) to demonstrate the effectiveness of the framework. Since these based prior floorplanning methods

works are fixed-outline, we compare our results with results reported by them, as the basic methods in our experiments.

For fair, the proposed and baseline methods should be having nearly the same area and other criteria. To clarify innovation of the proposed framework, we compare our experiments differently, because some methods have not experimental results on some criteria. In a next view, we aimed the comparisons with the three state-of-the-arts and recent floorplanning/placement algorithms only on reported wire-length results as Defer [36], EFP [37], G-TCG [35] and FNK [38].

5.3. Experimental Results Evaluation

In this section, we propose the experimental results in three different subsections. At first, our proposed framework (MFF) results are explained. At second, the convergence of our multi-pass SA-based method is explained. Finally, the analysis of the clustering method and Scalability of MFF is presented.

Empirical Results Analysis: The empirical results show the improvements in area, dead-space, wire length and running time by our framework respect to the baseline methods. The method proposed significant improvements in wire length, and runtime for all the benchmarks. Comparing to baseline methods, the proposed framework can reduce wire length by 23.2% on average and achieve 30X speed-up with scalable property. Our experiments are explained briefly as below.

Optimality and Applicability: A three-level convex optimization based framework is proposed for VLSI fixed-outline floorplanning design to solve an NP problem with a practical approach and reasonable running time (see table 6 and table 3). This is implemented based on a heuristic multi-pass SA method to catch speed and accurate global optimize floorplan result. This is combined with a new level as optimal clustering of blocks to present a best initial configuration to reduce the Wire length and running time of the problem (see table 3). Our accuracy on wire length is good and comparable to SAFFOA, UFO, LUO and FNK (see table 5 and figure 3). In this table, the reported FNK result about the ami49 and HP are optimized values [38]. The running time of our MFF is better (see table 4, table 5 and table 6).

Efficiency and Effectiveness: Our experimental results on MCNC and GSRC benchmarks are comparable with recent research on fixed-outline floorplanning. Our running time results are competitive on the MCNC with significant improvement on floorplan results. Also, our total wire length results are competitive on the MCNC and GSRC benchmarks with minimizing the chip size or the cost and impact on power minimization and delay which is the main objective of recent research floorplanning.

Scalability and Robustness: Our framework is scalable under fixed-die constraints. However, the classical approach is not scalable with fixed-outline floorplanning but our framework provides a greater improvement on wire length, running time (significant improved against UFO on increasing blocks, see table 4 and figure 4) and area with

increasing the number of blocks (see table 6). Our proposed framework guarantee min-dead space and complete area utilization in a fixed-outline floorplan with experiments on the original dimensions of the MCNC and GSRC benchmarks (see table 6). This produces aimed floorplan with any specified whitespace percentage. Some referenced works do not report their results or running time on complex ami33, ami49 and bigger such as n300 (see table 4).

Table 2. All properties of GSRC and MCNC benchmark suits

Circuit	Benchmark	No. of Blocks	No. of Connections	Area (mm ²)
n10	GSRC	10	27	2.46E-05
n30	GSRC	30	123	5.221E-05
n50	GSRC	50	255	13.81E-05
n100	GSRC	100	519	22.21E-05
n300	GSRC	300	1304	55.720E-07
apte	MCNC	9	21	4.781E-05
xerox	MCNC	10	35	4.84E-05
hp	MCNC	11	14	3.925E-06
ami33	MCNC	33	74	1.279E-05
ami49	MCNC	49	195	8.86136E-06

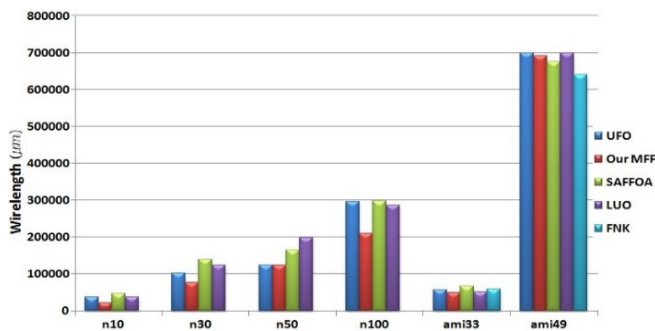
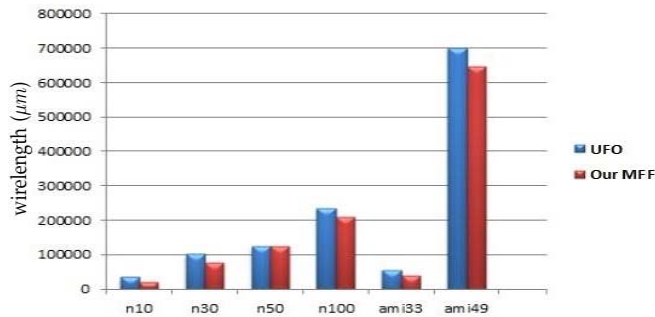
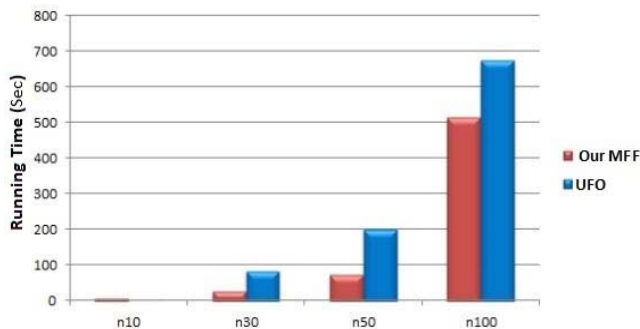


Figure 3. The wire length results of our MFF, SAFFOA, UFO, LUO and FNK



(a) The wire length results of our MFF and UFO



(b) The running time of our MFF and UFO

Figure 4. The wire length and running time results of our

Applicability and Flexibility: However, our framework experiments on the soft modules to minimize wire length with excellent results, it is applicable to the hard modules to minimize area (see table 7 and figure 5). However, we can divide the needed whitespace areas into special small blocks (or hard modules) and apply on our floorplanning process as special blocks. Also, the framework is applicable to classical methods for an optimization design.

Comparisons with the State-of-the-Art and Recent Algorithms:

Our framework experiments on the GSRC hard modules to minimize wire length with good results (see table 8). In table 8, the best results among the algorithms for individual benchmark test cases are shown in bold case where MFF shows the best results followed by EFP and Defer. Moreover, it is obvious that the wire length average results for all the algorithms are larger than the MFF. The results of MFF in table 8 show lower wire length, compared to EFP and Defer in the benchmark circuits n100a, n200a, and n300. This indicates that MFF s wire length results show the improvement in terms of wire length and lower dependency on aspect ratio as compared to EFP, Defer, and G-TCG with lower variance.

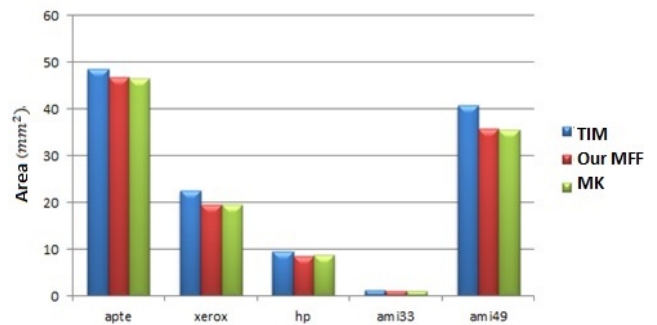


Figure 5. The area results of our MFF, TIM and MK

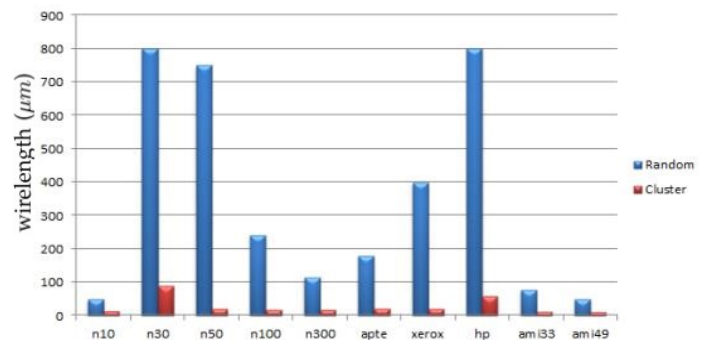
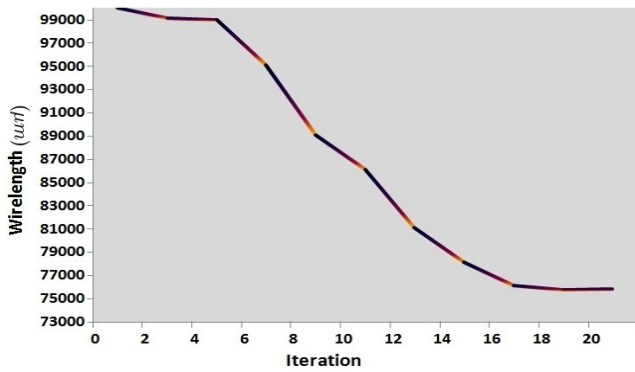


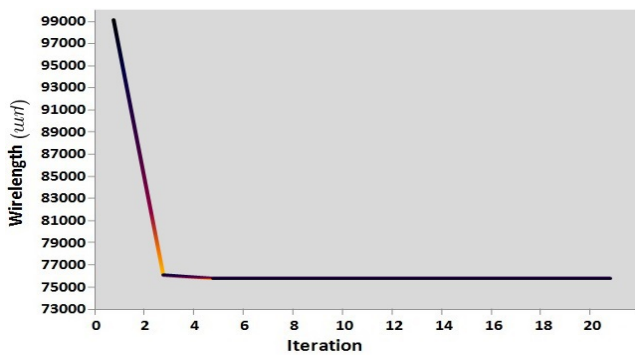
Figure 6. The wire length results of level two of our MFF Clustering and Random Distribution

Proposed MFF Convergence: In this section of experiments, we investigate the effectiveness of the block distribution (or clustering level with minimum connection cost optimization) on the algorithm reparation against a random initialization of the initial blocks to the floorplanning framework (see figure 7). The clustering level method can cluster all blocks with more connection in a cluster to optimize the minimum connection cost. The figure shows the speed convergence of the proposed framework (see figure

7(a)) is better than a random initialization framework (see figure 7(b)).



(a) Convergence of the Proposed Modeling with Random Initialization Blocks



(b) Convergence of the Proposed Modeling with our Clustering Level

Figure 7. Sample n30 circuit to convergence of our multi-pass SA method with clustering against a random block initialization for 20 iterations

Table 3. The wire length results of level two of our MFF Framework with Clustering and Random Distribution

Benchmark	Start with Clustering Method		Start with Random Distribution	
	First result (μm)	Optimal result (μm)	First result (μm)	Optimal result (μm)
n10	21	7	57	7
n30	62	3	803	3
n50	29	9	759	9
n100	22	4	244	4
n300	17	8	115	8
apte	33	11	191	11
Xerox	26	6	406	6
HP	106	46	864	46
ami33	21	8	86	8
ami49	21	10	60	10

Table 4. The wire length and running time results of our MFF and UFO

Benchmark	Our MFF		UFO	
	Wirelength (μm)	Running Time (Sec)	Wirelength (μm)	Running Time (Sec)
n10	21971	13	36398	3
n30	75833	31	102100	87
n50	123313	75.1	124300	204
n100	209283	515	296229	677
ami33	49426	34	55827	-
ami49	690509	126	699540	-

We see that our framework is a speed convergence approach with the lowest reparation on average with from 2 to 4 against from 18 to 20 on the random. The proposed

distribution can reduce the running time of Second Level of the proposed framework to reach the minimum wire length of its cost function. The results of the proposed framework on standard benchmark suits are presented in table 3. The Second Level results show improvements of wire length reduction (see figure 6) with using the Clustering Level.

Table 5. The wire length results of our MFF, SAFFOA, UFO, LUO and FNK

Benchmark	Wirelength (μm)				
	Our MFF	SAFFOA	UFO	LUO	FNK
n10	21971	46207	36398	37090	-
n30	75833	138218	102100	123860	-
n50	123313	165366	124300	197490	-
n100	209283	296229	296229	285070	-
ami33	49426	67364	55827	50081	58627
ami49	690509	675540	699540	699400	640509
apte	37018	-	-	-	404510
Xerox	16250	-	-	-	37099
HP	28303	-	-	-	15332

Table 6. The wire length and running time results of our MFF

Benchmark	Wirelength (μm)	Running Time (Sec)	Area (mm^2)	Deadspace (%)
n10	21971	13	22.21	0.2
n30	75833	31	21.29	0.5
n50	123313	75.1	20.73	4.4
n100	209283	515	19.96	11.24
ami33	49426	34	1.18	2.3
ami49	690509	126	35.75	0.99
apte	37018	8	46.96	0.88
Xerox	16250	3	19.52	0.9
HP	28303	1.5	8.5	3.1

Clustering Analysis: One issue during the implementation is that sometimes when using eigs () to compute eigenvectors, not all eigenvalues converge. Also, because the result of k-means largely depends on the initial condition, the solution of our algorithm is not deterministic. At the worst time, empty clusters will be formed. We find from this experiment that when the number of clusters is reduced, the HPWL decreases (i.e. floorplan quality improves).

This happens because smaller numbers of clusters allow the planner to perform better in terms of optimizing a floorplanning. We find first the result on partitioning, the graph has 4 connected components, and using k=4 to perform a cut algorithm, we successfully get the optimal partitioning solution with 0 cut. We also notice in the results that the percentage of overlap decreases when the number of clusters formed are relatively less.

For performing ratio cut on our benchmarks with the k set as 6, 8 and 16, the results show a great reduction of the total cuts compared with randomly picking edges as cuts in the benchmark graph. However, we find the size of ill-balanced partitioning is related to the graph topology. As we have stated, the graph contains only few connected components and the collaboration graph contains multiple uniformly distributed connected components.

The experiment results show that n cut can give a good approximation of the min-cut graph partitioning problem in terms of reducing the cut size. To cope the computational challenge of large scale clustering problem, for parallel computing benefits the eigenvector computing and k-means algorithm can be implemented on GPU.

Scalability Analysis: A SA-based approach is inspired from the foraging behavior of metal. The principal reasons of the inefficiency of SA-based algorithm in scalability are (1) the convergence closed to the optimum is strong due to the contributions of the best solution as the initial level are approximately the same, which means that values have significant influence on the block selection and running time. (2) The solution is clearly promising in getting good result and it belongs to initial point selection; this is because the considerable extension of the search space is created when the size of the problem is increasing. Adding convex modeling is very helpful on the best solution but time consuming. Our Multi-level SA-based meta-heuristic approach is a peaceful search technique. The principal reasons of the efficiency of our SA-based algorithm in scalability are:

- The best solution of the convex modeling is improved by breaking the concurrent objectives on different levels of the framework.
- The convergence closed to the optimum is strong due to the contributions of the best solution as the initial level is different, which means that clustering values have significant influence on the block selection and running time.
- The multi-level solution does not keep changing which are clearly promising in getting good results. This is because the considerable extension of the search space is not created and this issue is magnified when the size of the problem is increasing.
- When the cluster boundaries are resolved and blocks are assigned their absolute locations on the chip, clusters will no longer be present in memory.

6. Conclusion

A multi-level optimization SA-based framework was proposed to solve the fixed-outline floorplanning problem with rapid convergence. This is a global optimization problem for wire length minimization and total area utilization on a multi-pass framework. We applied a clustering level for initial distribution of the modules to reduce the running time of our framework. We made a scalable convex model to the original fixed-die floorplanning problem. The experimental results in the standard benchmarks demonstrate a scalable overlap-free and min dead space floorplans that obtain significant improvements on the wire length and running time with complete chip area utilization. This can be applied to variable-die with the proposed improvement in area as well as hard module case.

Our future works are concentrated on using blocks with different shape and reducing more running time. Next, we apply the method for whitespace distributions with hierarchical and recursively design of internal of each placed block to optimize the whitespace distribution with fixed placement of some blocks on the rectangular chip area to reduce the power and/or temperature of a chip. Another promising area for research would be to come up with a technique to predict the possible best shape for a cluster without actually placing it. This would improve the runtime of this technique a lot by eliminating the need to execute a floorplanner for multiple iterations. Also, the floorplanning of standard blocks inside all the possible shapes for a cluster

was done in parallel, and then it would also save a lot on runtime overhead. This would decrease the overall runtime requirements to a large extent. Using a non-linear based floorplanning technique which is faster compared to linear based approach is an area of research that promises to yield superior results.

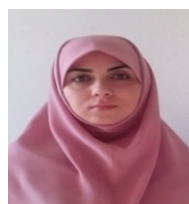
Acknowledgment

The authors acknowledge the High Performance Computing Center (HPC) of Institute for Research in Fundamental Sciences (IPM-Iran) to provide high performance computing resources that have contributed to the research results reported within this paper.

References

- [1] F. Brglez, "ACM/SIGDA Benchmark Electronic Newsletter DAC," Microelectronics Center of North Carolina (MCNC), 1993.
- [2] J. M. Lin, and Z. X. Hung, "UFO: Unified convex optimization algorithms for fixed-outline floorplanning considering pre-placed modules," IEEE Transactions Computer-Aided Design of Integrated Circuits and Systems, 30, 1034-1044, 2011.
- [3] C. Luo, M. F. Anjos, and A. Vannelli, "Large-scale fixed-outline floorplanning design using convex optimization techniques," Design Automation Conference, ASPDAC, Asia and South Pacific, 198-203, 2008.
- [4] J. Z. Yan, and C. Chu, "SDS: An Optimal Slack-Driven Block Shaping Algorithm for Fixed-Outline Floorplanning," IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, 32(2), 175-188, 2013.
- [5] D. S. Chen, C. T. Lin, Y.-W. Wang, and C. H. Cheng, "Fixed-outline floorplanning using robust evolutionary search," Engineering Applications of Artificial Intelligence, 20, 821-830, 2007.
- [6] O. He, S. Dong, J. Bian, S. Goto, and C. K. Cheng, "A novel fixed-outline floorplanner with zero deadspace for hierarchical design," IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 16-23, 2008.
- [7] S. N. Adya, and I. L. Markov. "Fixed-outline floorplanning: Enabling hierarchical design," IEEE Transaction Very Large Scale Integration, 11(6), 1120-1135, Dec. 2003.
- [8] A. B. Kahng, "Classical floorplanning harmful?," ISPD, 207-213, 2000.
- [9] JEDEC Solid State Technology Association, Failure Mechanisms and Models for Semiconductor Devices, JEDEC Publication JEP122E, Mar. 2009.
- [10] L. Stockmeyer, "Optimal orientations of cells in slicing floorplan designs, inform and control," Information Control, 57(2-3), 91-101, 1983.

- [11] A. Unutulmaz, G. Dndar, and FV. Fernndez, "On the convex formulation of area for slicing floorplans," *Integration, the VLSI Journal*, 50, 74-80, 2015.
- [12] R. Viswanath, V. Wakharkar, A. Watew, and V. Lbonheur, "Thermal performance challenges from silicon to systems," *Intel Technology Journal*, 4(3), 1-16, 2000.
- [13] International technology roadmap for semiconductors, 2010, <http://www.itrs.net/Links/2010ITRS/Home2010.htm>.
- [14] A. Kennings, and I. Markov, "Analytical minimization of half-perimeter wire-length," *Proceeding of Asia and South Pacific Design Automation Conference*, 179-184, 2000.
- [15] H. Murata, and E. S. Kuh, "Sequence-pair based placement method for hard/soft/pre-placed modules," *Proceedings of the international symposium on Physical design*, 167-172, 1998.
- [16] J. Cong, T. Kong, D. Xu, F. Liang, J. S. Liu, and W. H. Wong, "Relaxed simulated tempering for VLSI floorplan designs," *Design Automation Conference, Proceedings of the ASP-DAC'99, Asia and South Pacific*, 13-16, 1999.
- [17] T. C. Chen, Y. W. Chang, and S. C. Lin, "A new multilevel framework for large-scale interconnect-driven floorplanning," *IEEE Transaction Computer-Aided Design, Integrated Circuits System*, 27(2), 286-294, 2007.
- [18] GigabyteSystemsResearchCenter[online]. Available at: <http://www.cse.ucsc.edu/research/surf/GSRC/progress.html>.
- [19] Microelectronics Center of North Carolina [online]. Available at: <http://www.lyle.smu.edu/manikas/Benchmarks/MCNC-Benchmark-Netlists.html>.
- [20] R. Foraita, J. Spallek, and H. Zeeb, "Directed Acyclic Graphs," *Handbook of Epidemiology*, Springer, 1481-1517, 2014.
- [21] T. Nakul, "LCPlace: A novel VLSI placement methodology based on large cluster formation," MS thesis, University of Cincinnati, 2014.
- [22] Y. Zhan, Y. Feng, and S. Sapatnekar, "A fixed-die floorplanning algorithm using an analytical approach," *Proceeding ASP-DAC*, 771-776, 2006.
- [23] E. K. P. Chong, and S. H. Zak, "An Introduction to Optimization," 2nd ed. NewYork: Wiley, 2001.
- [24] A. Unutulmaz, G. Dundar, and F.V. Fernandez, "Area optimization on fixed analog floorplans using convex area functions," *Proceedings of the Design, Automation and Test in Europe Conference, Grenoble*, 1843-1848, 2013.
- [25] M. Sarrafzadeh, and C. K. Wong, "An Introduction to VLSI Physical Design," McGraw-Hill Higher Education, 1996.
- [26] S. I. Dhillon, G. Yuqiang, and K. Brian, "Kernel k-means: spectral clustering and normalized cuts," *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD04*, 551-556, 2004.
- [27] A. Y. Ng, M. I. Jordan, W. Yair, and et Al., "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, 2, 849-856, 2002.
- [28] L. Jin, D. Kim, L. Mu, D. S. Kim, and S. M. Hu, "A sweepline algorithm for Euclidean Voronoi diagram of circles," *IEEE Transaction on Computer-Aided Design*, 38(3), 260-272, Mar. 2006.
- [29] B. Sowmya, and M. Sunil, "Minimization of Floorplanning Area and Wire Length Interconnection Using Particle Swarm Optimization," *International Journal of Emerging Technology and Advanced Engineering*, 3, 2013.
- [30] M. B. Healy, M. Vittes, M. Ekpanyapong, C. S. Ballapuram, S. K. Lim, H. -H. S. Lee, and G. H. Loh, "Multiobjective microarchitectural floorplanning for 2-D and 3-D ICs," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 26(1), 38-52, 2007.
- [31] E. K. P. Chong, and S. H. Zak, "An Introduction to Optimization," 2nd Web page of <http://thefreedictionary.com> for number plates regulations of many countries.



Houra Banihashem received his B.Sc. in Computer Science from Sharif University of Technology, Tehran, Iran and M.Sc. in Computer Architecture from Shahed University, Tehran, Iran. Her research interests are Computer Architecture, VLSI Design and Graph

Theory.

E-mail: hbanihashem@gmail.com



Aminollah Mahabadi is currently an Assistant Professor in the Electrical and Computer Engineering Department, Shahed University, Tehran, Iran and a researcher in School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Iran. His research interests include Network on a Chip (NoC), Intelligent Modeling and Simulation, Parallel Processing, Social Networks and Distributed Systems.

E-mail: mahabadi@shahed.ac.ir

Paper Handling Data:

Submitted: 13.03.2015

Received in revised form: 19.12.2015

Accepted: 02.02.2016

Corresponding author: Dr. Aminollah Mahabadi,
Department of Electrical Engineering, Shahed
University, Tehran, Iran.